

## 项目 3 配置网络和使用 ssh 服务-实训任务指导书 (3-1)

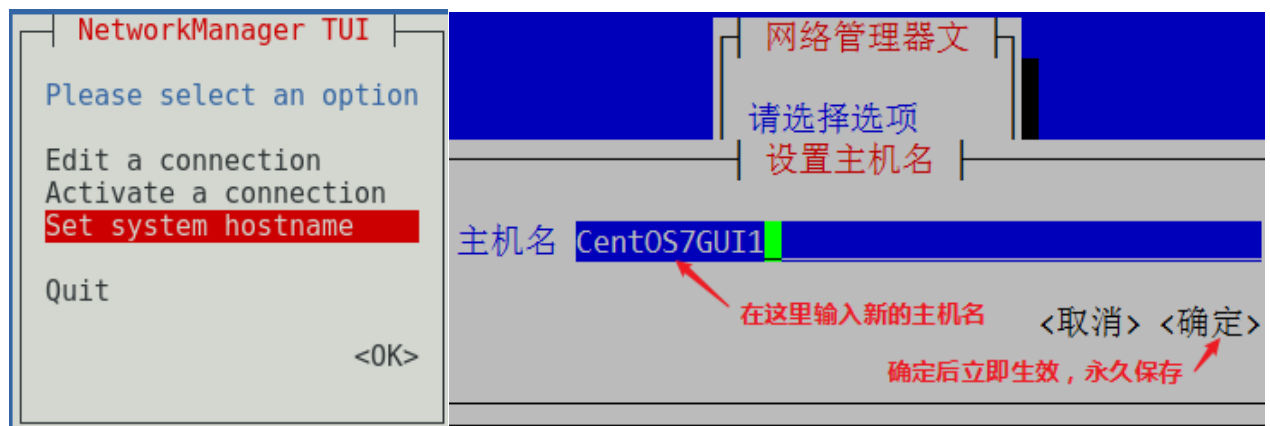
任务 1: 请分别使用 nmtui、hostnamectl 和 nmcli 三种不同的方法, 设置修改主机名。

### 方法一: 使用 nmtui 修改主机名

nmtui 提供基于光标的文本用户界面 (TUI), nmcli 是命令行的 NetworkManager 工具, 自动把配置写到 /etc/sysconfig/network-scripts/ 目录下面。nmtui 命令会开启一个图形化的配置工具, 通过 nmtui 提供的 GUI 界面, 可以编辑连接, 启动连接, 设置主机名。可以使用箭头键或按 Tab 键向前选择选项, 按 Shift+Tab 组合键返回。按 Enter 选择一个选项。按 Space 键选择复选框状态。

1. [root@RHEL7-1 ~]# nmtui

在如下图所示的界面中进行配置 (TAB 键和上下方向键配合使用进行选择)



### 2. 重启 hostnamed 服务

使用 NetworkManager 的 nmtui 接口修改了静态主机名后 (/etc/hostname 文件), 不会通知 hostnamectl。要想强制让 hostnamectl 知道静态主机名已经被修改, 需要重启 hostnamed 服务。

```
[root@RHEL7-1 ~]# systemctl restart systemd-hostnamed
```

3. 可使用 `hostname` 或 `hostnamectl status` 命令查看修改后的情况。

```
[root@192 ~]# hostname
CETOS7GUI1
[root@192 ~]# hostnamectl status
  Static hostname: CETOS7GUI1
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 9d5690126a88460f898b3a3ff5cf1c59
        Boot ID: 7d33116e2601442fa5b48a411d66ca61
  Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
        CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-1160.el7.x86_64
  Architecture: x86-64
```

## 方法二：使用 `hostnamectl` 修改主机名

(1) 查看主机名

```
[root@RHEL7-1 ~]# hostnamectl status
```

```
  Static hostname: Centos7GUI1
```

```
  Pretty hostname: Centos7GUI1
```

(2) 设置新的主机名

```
[root@RHEL7-1 ~]# hostnamectl set-hostname my.smile.com
```

(3) 查看主机名

```
[root@RHEL7-1 ~]# hostnamectl status
```

```
  Static hostname: my.smile.com
```

**说明：**1.临时修改主机名：`hostname` 临时主机名，不对 `/etc/hostname` 文件的内容进行修改。

2.永久修改主机名：`hostnamectl set-hostname` 永久主机名 是对 `/etc/hostname` 文件的内容进行修改。

### 方法三：使用 NetworkManager 的命令行接口 nmcli 修改主机名

nmcli 可以修改/etc/hostname 中的静态主机名。

#### 1. 查看主机名

```
[root@RHEL7-1 ~]# nmcli general hostname  
my.smile.com
```

#### 2. 设置新主机名

```
[root@RHEL7-1 ~]# nmcli general hostname Centos7-1  
[root@RHEL7-1 ~]# nmcli general hostname  
Centos7-1
```

#### 3. 重启 hostnamed 服务让 hostnamectl 知道静态主机名已经被修改

```
[root@RHEL7-1 ~]# systemctl restart systemd-hostnamed
```

```
[root@192 ~]# nmcli general hostname  
my.smile.com  
[root@192 ~]# nmcli general hostname Centos7-1  
[root@192 ~]# nmcli general hostname  
Centos7-1  
[root@192 ~]# vim /etc/hostname  
[root@192 ~]# hostnamectl status  
Static hostname: Centos7-1  
Icon name: computer-vm  
Chassis: vm  
Machine ID: 9d5690126a88460f898b3a3ff5cf1c59  
Boot ID: 7d33116e2601442fa5b48a411d66ca61  
Virtualization: vmware  
Operating System: CentOS Linux 7 (Core)  
CPE OS Name: cpe:/o:centos:centos:7  
Kernel: Linux 3.10.0-1160.el7.x86_64  
Architecture: x86-64  
[root@192 ~]# systemctl restart systemd-hostnamed
```

#### 说明：

也可以使用修改配置文件 vim /etc/hostname, 后需要重启 hostnamed 服务生效。

**任务 2：请分别使用系统菜单、网卡配置文件、图形界面和 nmcli 命令四种不同的方法来配置网络。**

准备工作：

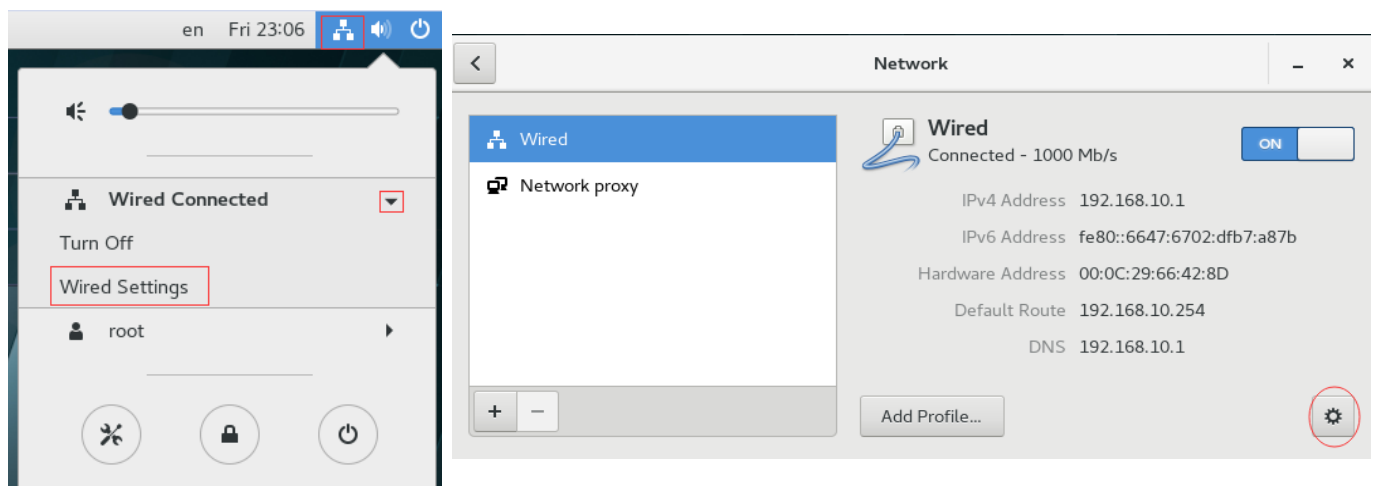
可恢复到快照 1 状态。



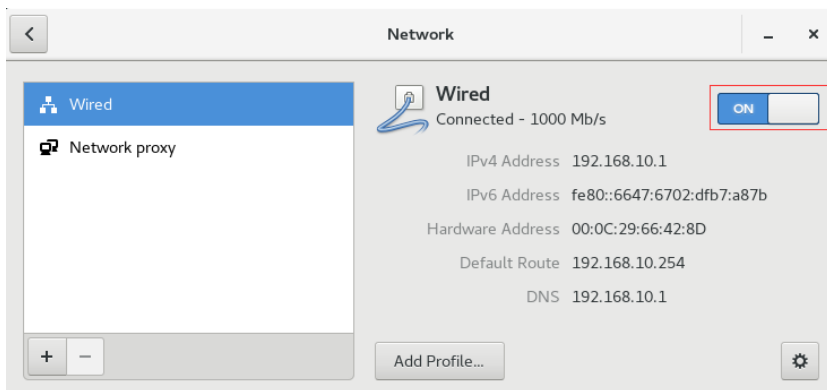
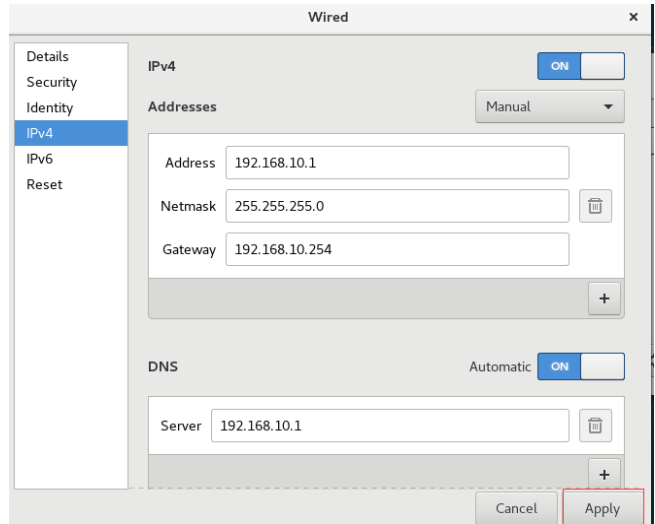
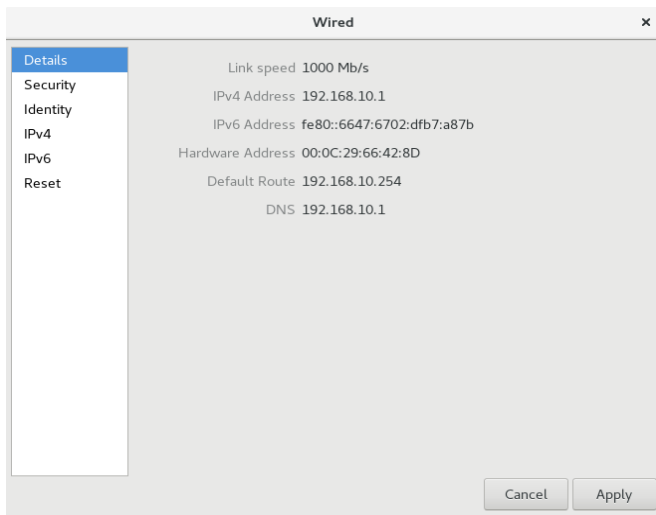
### 方法一：使用系统菜单配置网络（首选方法）

在 Linux 系统上配置服务之前，必须先保证主机之间能够顺畅地通信。

可以单击桌面右上角的网络连接图标  打开网络配置界面，一步步完成网络信息查询和网络配置。



配置完成后，单击“Apply”按钮应用配置回到图 2-9 所示的界面。注意网络连接应该设置在“ON”状态，如果在“OFF”状态，请进行修改。注意，有时需要重启系统配置才能生效。



可使用命令：**ip add show** 查看。

使用 firefox <http://192.168.10.1> 访问成功。

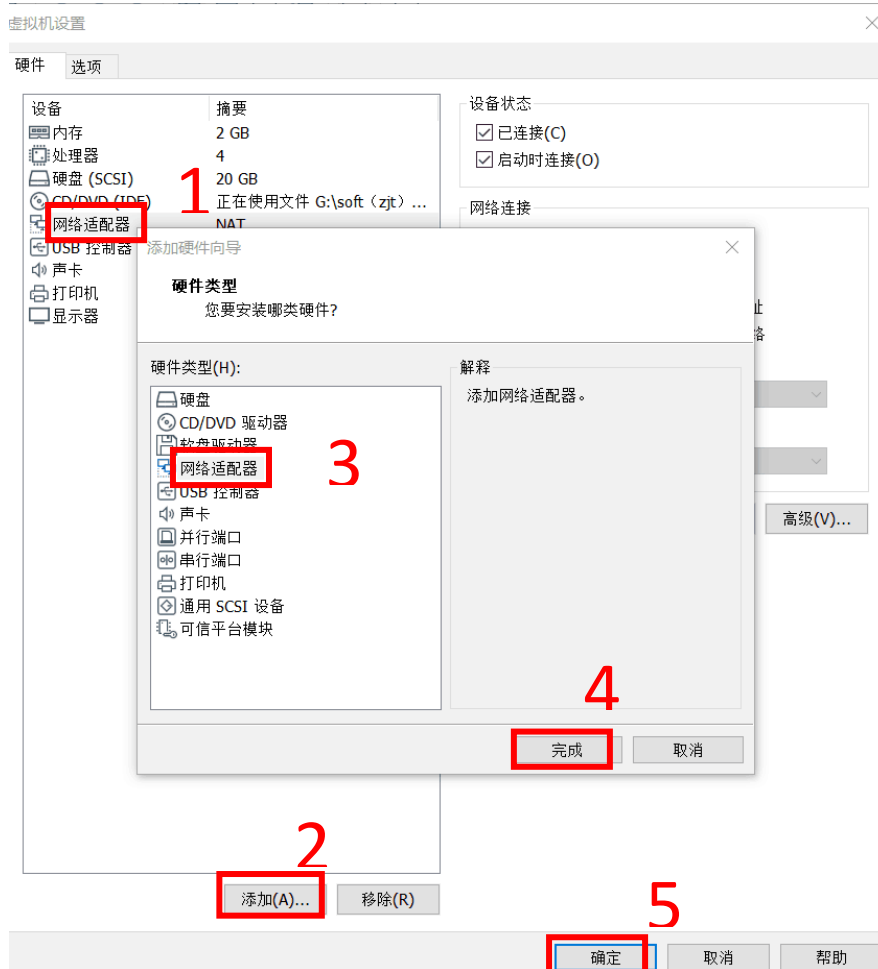


## 方法二：通过网卡配置文件配置网络（不建议使用）

在 RHEL 7 中，网卡配置文件的前缀则以 ifcfg 开始，如 ifcfg-ens37。

名称为 ifcfg-ens37 的网卡设备，将其配置为开机自启动，并且 IP 地址、子网、网关等信息由人工指定，其步骤如下。

准备工作：**在虚拟系统里，先添加一块新网卡：**



```
[root@192 ~]# ifconfig
ens33: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.58.165 netmask 255.255.255.0 broadcast 192.168.58.255
    inet6 fe80::b9bc:46be:e31d:dd38 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 17399 bytes 24900237 (23.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8848 bytes 552302 (539.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.58.167 netmask 255.255.255.0 broadcast 192.168.58.255
    inet6 fe80::506e:dd3d:4f66:5db1 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:3c txqueuelen 1000 (Ethernet)
    RX packets 22 bytes 4243 (4.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 4630 (4.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

(1) 使用 `cd` 命令切换到 `/etc/sysconfig/network-scripts` 目录中（存放着网卡的配置文件）。

(2) 使用 `vim` 编辑器修改网卡文件 `ifcfg-ens37`，逐项写入下面的配置参数并保存退出。由于每台设备的硬件及架构是不一样的，所以请读者使用 `ifconfig` 命令自

行确认各自网卡的默认名称。

```
设备类型: TYPE=Ethernet
地址分配模式 (两种): BOOTPROTO=static (静态) 或 BOOTPROTO=dhcp (动态)
网卡名称: NAME=ens37
是否启动: ONBOOT=yes
IP 地址: IPADDR=192.168.10.1
子网掩码: NETMASK=255.255.255.0
网关地址: GATEWAY=192.168.10.1
DNS 地址: DNS1=192.168.10.1
```

(3) 重启网络服务并测试网络是否联通。

进入到网卡配置文件所在的目录，然后编辑网卡配置文件，在其中填入下面的信息（不需要输入汉字）：可使用 `nmcli con show` 或 `nmcli con` 命令查看，从系统所有网卡的 UUID 中查看新添加网卡的 UUID。

```
[root@RHEL7-1 ~]# cd /etc/sysconfig/network-scripts/
```

```
[root@RHEL7-1 network-scripts]# vim ifcfg-ens37
```

```
TYPE=Ethernet           # 网卡类型: 为以太网
PROXY_METHOD=none       # 代理方式: 关闭状态
BROWSER_ONLY=no        # 只是浏览器: 否
BOOTPROTO=static       # 设置网卡获得 ip 地址的方式, 可能的选项为 static (静态), dhcp (dhcp 协议) 或
bootp (bootp 协议)
IPV4_FAILURE_FATAL=no   # 是不开启 IPV4 致命错误检测: 否
NAME=ens37
DEFROUTE=NO             # 默认路由: 否 IPV6_AUTOCONF=yes           # IPV6 是否自动配置:
是[不会有任何影响, 现在还没用到 IPV6]
UUID=9d5c53ac-93b5-41bb-af37-4908ccea6dc31
DEVICE=ens37
ONBOOT=yes
IPADDR=192.168.10.1
NETMASK=255.255.255.0
GATEWAY=192.168.10.1
DNS1=192.168.10.1
```

**重点配置如图所示的几项参数：（UUID 参数根据自己实际情况做相应改变）**



执行重启网卡设备的命令（在正常情况下不会有提示信息），然后通过 ping 命令测试网络能否联通。由于在 Linux 系统中 ping 命令不会自动终止，所以需要手动按下“Ctrl+C”组合键来强行结束进程。或 ping 命令使用 -c 选项，指定条数目的包后停止。如：ping -c 4 192.168.10.1 命令。

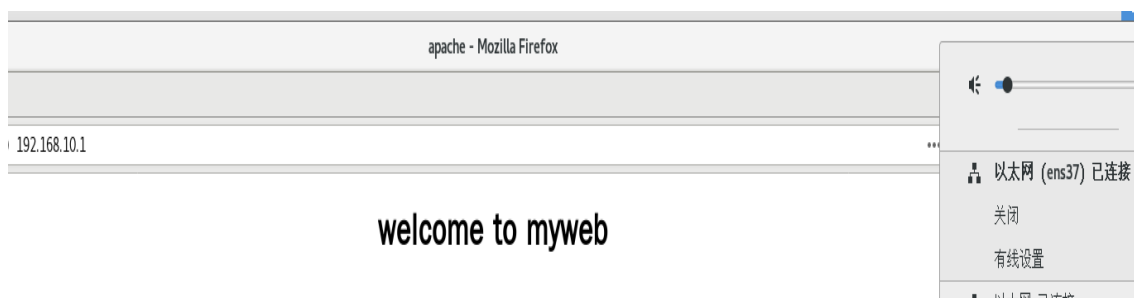
```
[root@RHEL7-1 network-scripts]# systemctl restart network
```

```
[root@RHEL7-1 network-scripts]# ping -c 4 192.168.10.1
```

```
rtt min/avg/max/mdev = 0.044/0.050/0.107/0.025 ms
[ root@192 network-scripts]# ping -c 4 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data:
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.060 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.049 ms

--- 192.168.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.049/0.055/0.060/0.008 ms
[ root@192 network-scripts]# █
```

使用 firefox <http://192.168.10.1> 访问成功，如下图。





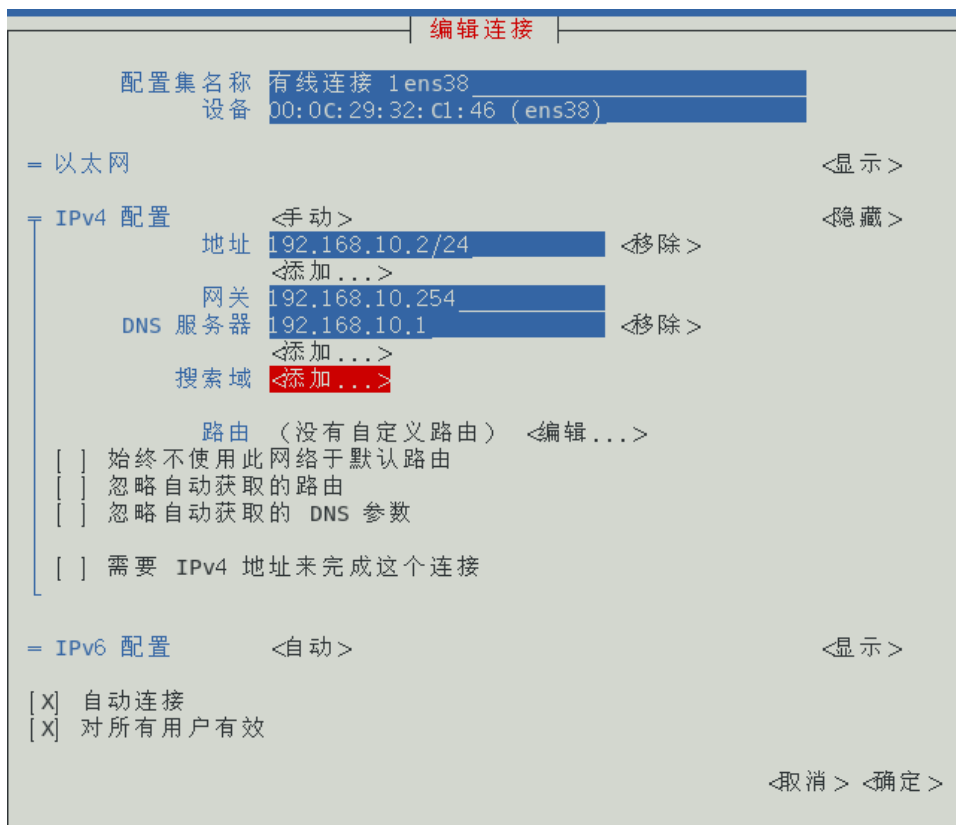
### 方法三： 使用图形界面配置网络

#### 准备工作： 添加新网卡 ens38

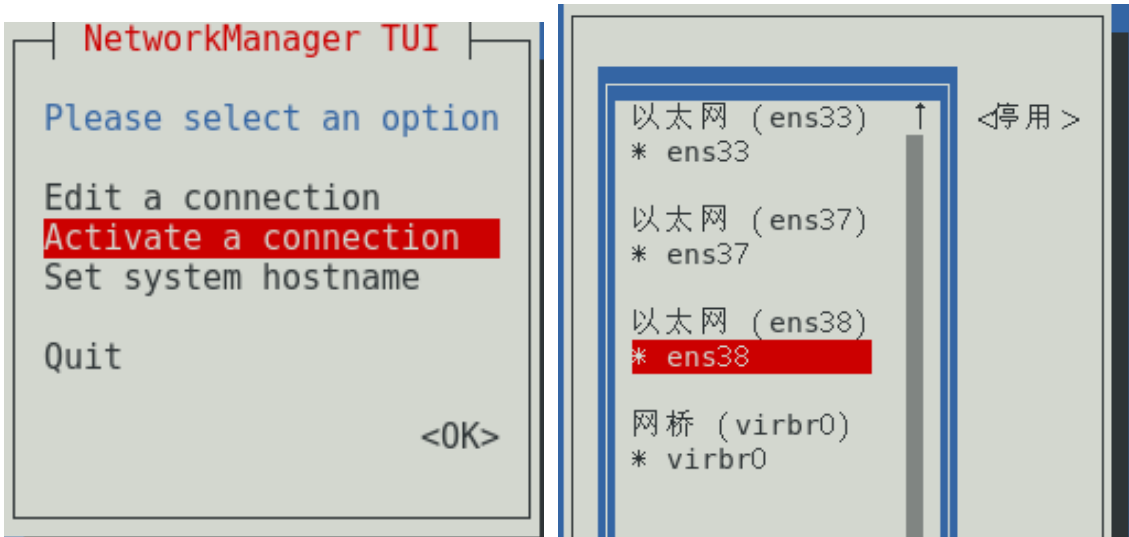
- (1) 使用 nmtui 命令来配置网络。
- (2) 显示图所示的图形配置界面。
- (3) 配置过程如下图所示。



(4) 按下“Show”（显示）按钮，显示信息配置框，如图 2-13 所示。在服务器主机的网络配置信息中填写 IP 地址 192.168.10.1/24 等信息，单击“OK”按钮，如图所示。



(5) 按 “<back>” 按钮回到 nmtui 图形界面初始状态，选中 “Activate a connection” 选项，激活刚才的连接 “ens38”。前面有 “\*” 号表示激活，如图所示。



(6) 使用 ifconfig 命令查看配置结果。

```
[root@192 network-scripts]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.58.165 netmask 255.255.255.0 broadcast 192.168.58.255
    inet6 fe80::b9bc:46be:e31d:dd38 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 109127 bytes 156790615 (149.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 53797 bytes 3423899 (3.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::20c:29ff:fe32:c13c prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:3c txqueuelen 1000 (Ethernet)
    RX packets 101 bytes 16944 (16.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 76 bytes 9685 (9.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.2 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::1588:7ce9:5461:133a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:46 txqueuelen 1000 (Ethernet)
    RX packets 86 bytes 12911 (12.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74 bytes 11161 (10.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@192 network-scripts]# ping -c 4 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data:
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.117 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.131 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.059 ms
```

(7) 使用 firefox <http://192.168.10.2> 访问成功，如下图。



## 方法四：使用 nmcli 命令配置网络

NetworkManager 是管理和监控网络设置的守护进程，设备即网络接口，连接是对网络接口的配置。一个网络接口可以有多个连接配置，但同时只有一个连接配置生效。

### 1. 创建新连接配置

(1) 创建新连接配置 default，IP 通过 DHCP 自动获取

```
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
```

```
[root@RHEL7-1 ~]# nmcli connection add con-name default type
```

Ethernet ifname ens38

连接 "default" (2ba76bd4-fcce-4353-9d07-9b4f582d4aa2) 已成功添加

```
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
default   2ba76bd4-fcce-4353-9d07-9b4f582d4aa2 ethernet  --
[root@192 network-scripts]# nmcli connection add con-name default type Ethernet ifname ens38
连接 "default" (2ba76bd4-fcce-4353-9d07-9b4f582d4aa2) 已成功添加。
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
default   2ba76bd4-fcce-4353-9d07-9b4f582d4aa2 ethernet  --
```

(2) 删除连接

```
[root@RHEL7-1 ~]# nmcli connection delete default
```

成功删除连接 “default” (2ba76bd4-fcce-4353-9d07-9b4f582d4aa2)。

```
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
default   2ba76bd4-fcce-4353-9d07-9b4f582d4aa2 ethernet  --
[root@192 network-scripts]# nmcli connection delete default
成功删除连接 "default" (2ba76bd4-fcce-4353-9d07-9b4f582d4aa2)。
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
[root@192 network-scripts]#
```

(3) 创建新的连接配置 test2, 指定静态 IP, 不自动连接

```
[root@RHEL7-1 ~]# nmcli connection add con-name test2 ipv4.method
manual ifname ens38 autoconnect no type Ethernet ipv4.addresses
192.168.10.100/24 gw4 192.168.10.3
```

连接 “test2” (b32e16a6-b327-4cba-9c13-fa650ae9287a) 已成功添加。

```
[root@192 network-scripts]# nmcli connection add con-name test2 ipv4.method manual ifname ens38 autoconnect no type Ethernet ipv4.addresses 192.168.10.100/24 gw4 192.168.10.3
连接 "test2" (b32e16a6-b327-4cba-9c13-fa650ae9287a) 已成功添加。
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
test2     b32e16a6-b327-4cba-9c13-fa650ae9287a ethernet  --
```

#### (4) 参数说明

- con-name: 指定连接名字, 没有特殊要求。
- ipv4.methmod: 指定获取 IP 地址的方式。
- ifname: 指定网卡设备名, 也就是次配置所生效的网卡。
- autoconnect: 指定是否自动启动。
- ipv4.addresses: 指定 IPv4 地址。
- gw4: 指定网关。

#### 2. 查看/etc/sysconfig/network-scripts/目录

```
[root@RHEL7-1 ~]# ls /etc/sysconfig/network-scripts/ifcfg-*
/etc/sysconfig/network-scripts/ifcfg-ens33
/etc/sysconfig/network-scripts/ifcfg-test2
```

```
/etc/sysconfig/network-scripts/ifcfg-lo
```

### 3. 启用 test2 连接配置

```
[root@RHEL7-1 ~]# nmcli connection up test2
```

连接已成功激活 (D-Bus 活动路径:

```
/org/freedesktop/NetworkManager/ActiveConnection/11)
```

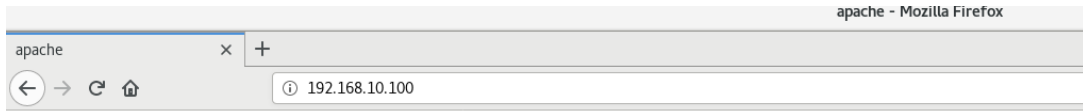
```
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
test2     b32e16a6-b327-4cba-9c13-fa650ae9287a ethernet  --
[root@192 network-scripts]# nmcli connection up test2
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/11)
[root@192 network-scripts]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
ens37     9d5c53ac-93b5-41bb-af37-4908cce6dc31 ethernet  ens37
test2     b32e16a6-b327-4cba-9c13-fa650ae9287a ethernet  ens38
virbr0    b29c356d-a026-4665-9f70-d258cc6455ff bridge    virbr0
ens38     b245d029-a534-337f-9619-9468eaca8ffe ethernet  --
```

### 4. 查看是否生效

```
[root@RHEL7-1 ~]# nmcli device show ens38
```

```
[root@192 network-scripts]# nmcli device show ens38
GENERAL.DEVICE:                ens38
GENERAL.TYPE:                  ethernet
GENERAL.HWADDR:                00:0C:29:32:C1:46
GENERAL.MTU:                   1500
GENERAL.STATE:                 100 (已连接)
GENERAL.CONNECTION:           test2
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/ActiveConnection/11
WIRED-PROPERTIES.CARRIER:    开
IP4.ADDRESS[1]:               192.168.10.100/24
IP4.GATEWAY:                   192.168.10.3
IP4.ROUTE[1]:                 dst = 192.168.10.0/24, nh = 0.0.0.0, mt = 104
IP4.ROUTE[2]:                 dst = 0.0.0.0/0, nh = 192.168.10.3, mt = 104
IP6.ADDRESS[1]:               fe80::26a4:ccfc:5ce1:f69d/64
IP6.GATEWAY:                   --
IP6.ROUTE[1]:                 dst = fe80::/64, nh = ::, mt = 104
IP6.ROUTE[2]:                 dst = ff00::/8, nh = ::, mt = 256, table=255
[root@192 network-scripts]#
```

5. 基本的 IP 地址配置成功, 使用 firefox <http://192.168.1.100> 访问成功。



welcome to myweb

## 6. 修改连接设置

(1) 修改 test2 为自动启动

```
[root@RHEL7-1 ~]# nmcli connection modify test2
```

```
connection.autoconnect yes
```

(2) 修改 DNS 为 192.168.10.1

```
[root@RHEL7-1 ~]# nmcli connection modify test2 ipv4.dns 192.168.10.1
```

(3) 添加 DNS 114.114.114.114

```
[root@RHEL7-1 ~]# nmcli connection modify test2 +ipv4.dns  
114.114.114.114
```

(4) 看下是否成功

```
[root@RHEL7-1 ~]# cat /etc/sysconfig/network-scripts/ifcfg-test2
```

```
[root@192 network-scripts]# cat /etc/sysconfig/network-scripts/ifcfg-test2  
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none  
IPADDR=192.168.10.100  
PREFIX=24  
GATEWAY=192.168.10.3  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no  
IPV6_INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no  
IPV6_ADDR_GEN_MODE=stable-privacy  
NAME=test2  
UUID=b32e16a6-b327-4cba-9c13-fa650ae9287a  
DEVICE=ens38  
ONBOOT=yes  
DNS1=192.168.10.1  
DNS2=114.114.114.114  
[root@192 network-scripts]#
```

(5) 删除 DNS

```
[root@RHEL7-1 ~]# nmcli connection modify test2 -ipv4.dns 114.114.114.114
```

(6) 修改 IP 地址和默认网关

```
[root@RHEL7-1 ~]# nmcli connection modify test2 ipv4.addresses  
192.168.10.200/24 gw4 192.168.10.254
```

(7) 还可以添加多个 IP

```
[root@RHEL7-1 ~]# nmcli connection modify test2 +ipv4.addresses  
192.168.10.250/24
```

```
[root@192 network-scripts]# cat /etc/sysconfig/network-scripts/ifcfg-test2  
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none  
IPADDR=192.168.10.200  
PREFIX=24  
GATEWAY=192.168.10.254  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no  
IPV6_INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no  
IPV6_ADDR_GEN_MODE=stable-privacy  
NAME=test2  
UUID=b32e16a6-b327-4cba-9c13-fa650ae9287a  
DEVICE=ens38  
ONBOOT=yes  
DNS1=192.168.10.1  
IPADDR1=192.168.10.250  
PREFIX1=24  
[root@192 network-scripts]# █
```

(8) 如下图网络配置使用 test2，使用 firefox <http://192.168.1.200> 和 firefox <http://192.168.1.250> 均访问成功。



```
[root@RHEL7-1 ~]# nmcli connection show "test2"
```

```
[root@192 network-scripts]# nmcli connection show "test2"
connection.id: test2
connection.uuid: b32e16a6-b327-4cba-9c13-fa650ae9287a
connection.stable-id: --
connection.type: 802-3-ethernet
connection.interface-name: ens38
connection.autoconnect: 是
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.multi-connect: 0 (default)
connection.auth-retries: -1
connection.timestamp: 1648055320
connection.read-only: 否
connection.permissions: --
connection.zone: --
connection.master: --
connection.slave-type: --
connection.autoconnect-slaves: -1 (default)
connection.secondaries: --
connection.gateway-ping-timeout: 0
connection.metered: 未知
connection.lldp: default
connection.mdns: -1 (default)
connection.llmnr: -1 (default)
802-3-ethernet.port: --
802-3-ethernet.speed: 0
802-3-ethernet.duplex: --
802-3-ethernet.auto-negotiate: 否
802-3-ethernet.mac-address: --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.generate-mac-address-mask: --
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu: 自动
802-3-ethernet.s390-subchannels: --
802-3-ethernet.s390-nettype: --
802-3-ethernet.s390-options: --
802-3-ethernet.wake-on-lan: default
802-3-ethernet.wake-on-lan-password: --
ipv4.method: manual
ipv4.dns: 192.168.10.1
ipv4.dns-search: --
lines 1-40
```



### 任务 3-3 创建网络会话实例：

如果我们在公司网络中使用笔记本电脑时需要手动指定网络的 IP 地址，而回到家中则是使用 DHCP 自动分配 IP 地址。这就需要麻烦地频繁修改 IP 地址，但是使用了网络会话功能后一切就简单多了——只需在不同的使用环境中激活相应的网络会话，就可以实现网络配置信息的自动切换了。假设将公司网络中的网络会话称之为 **company**，将家庭网络中的网络会话称之为 **home**，依次创建 nmcli 命令各自的网络会话。

nmcli 是一款基于命令行的网络配置工具，功能丰富，参数众多。它可以轻松地查看网络信息或网络状态：

#### 实训步骤：

```
[root@RHEL7-1 ~]# nmcli connection show
```

如下图：

```
[root@192 ~]# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
ens33	64d772b6-4d24-4534-9ce0-b8efd2e25664	ethernet	ens33
virbr0	b58c1df5-8ffb-4ff5-b7ed-0f6c6be983d3	bridge	virbr0

#### 实训步骤：

(1) 使用 con-name 参数指定公司所使用的网络会话名称 company，然后依次用 ifname 参数指定本机的网卡名称。用 autoconnect no 参数设置该网络会话默认不被自动激活，以及用 ip4 及 gw4 参数手动指定网络的 IP 地址：

```
[root@RHEL7-1 ~]# nmcli connection add con-name company ifname ens33  
autoconnect no type ethernet ip4 192.168.10.1/24 gw4 192.168.10.254
```

如下图：

```
[root@192 ~]# nmcli connection add con-name company ifname ens33 autoconnect no
type ethernet ip4 192.168.10.1/24 gw4 192.168.10.254
连接 "company" (ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a) 已成功添加。
```

(2) 使用 con-name 参数指定家庭所使用的网络会话名称 home。我们想从外部 DHCP 服务器自动获得 IP 地址，因此这里不需要进行手动指定。

```
[root@RHEL7-1 ~]# nmcli connection add con-name home type ethernet ifname
ens33
```

如下图：

```
[root@192 ~]# nmcli connection add con-name home type ethernet ifname ens33
连接 "home" (8be600db-228b-42d2-a581-7f8893413efa) 已成功添加。
```

(3) 在成功创建网络会话后，可以使用 nmcli 命令查看创建的所有网络会话：

```
[root@RHEL7-1 ~]# nmcli connection show
```

如下图：

```
[root@192 ~]# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
ens33	64d772b6-4d24-4534-9ce0-b8efd2e25664	ethernet	ens33
virbr0	b58c1df5-8ffb-4ff5-b7ed-0f6c6be983d3	bridge	virbr0
company	ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a	ethernet	--
home	8be600db-228b-42d2-a581-7f8893413efa	ethernet	--

(4) 使用 nmcli 命令配置过的网络会话是永久生效的，这样当我们下班回家后，顺手启用 home 网络会话，网卡就能自动通过 DHCP 获取到 IP 地址了。

```
[root@RHEL7-1 ~]# nmcli connection up home
```

如下图：

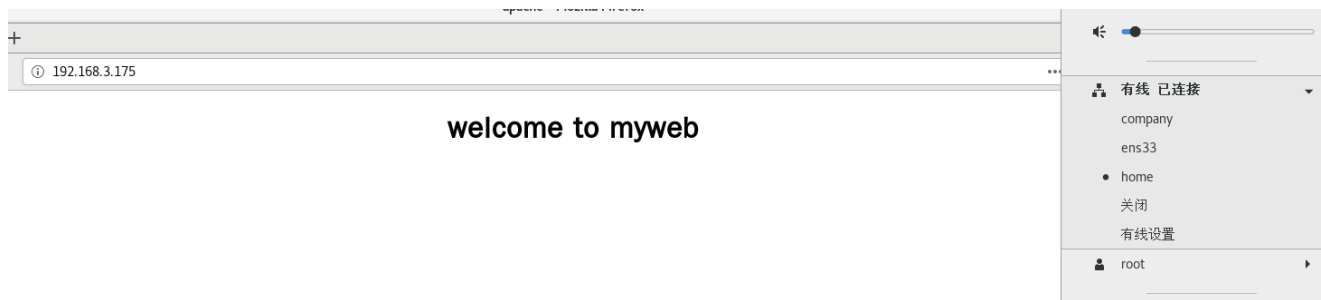
```
[root@192 ~]# nmcli connection up home
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/4)
```

```
[root@RHEL7-1 ~]# ifconfig
```

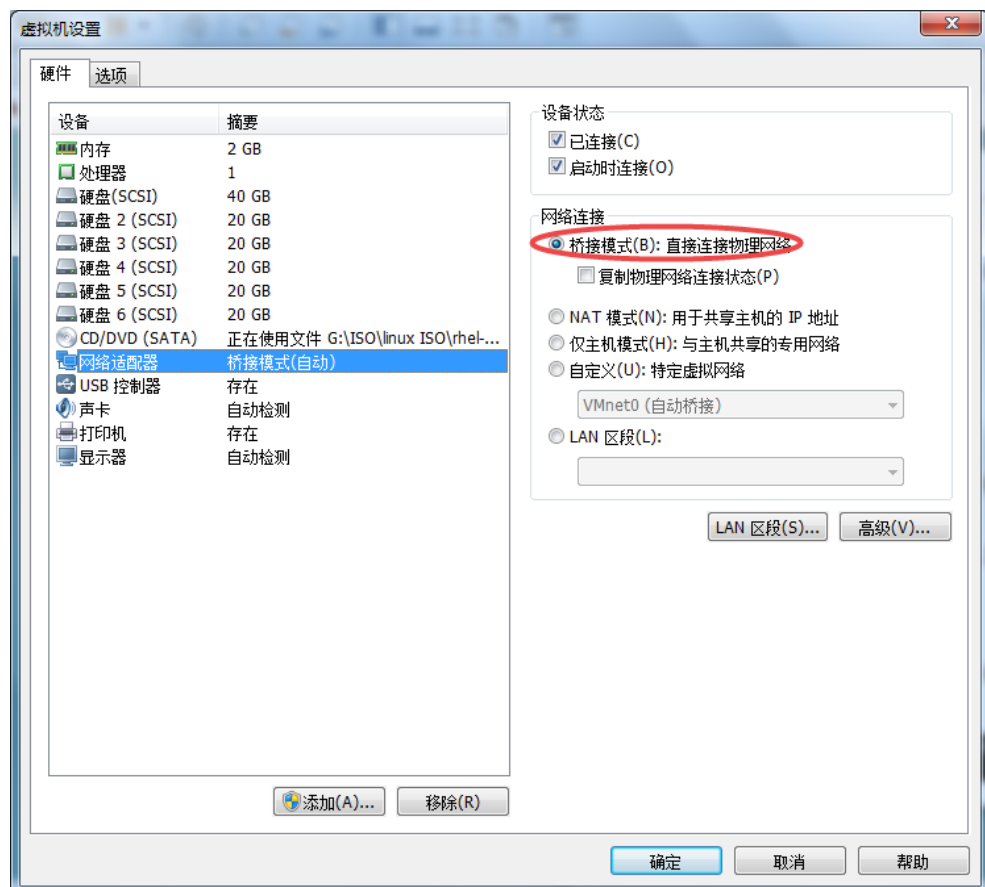
如下图：

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.3.175 netmask 255.255.255.0 broadcast 192.168.3.255
    inet6 fe80::9a9e:2464:65e2:b15f prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 413 bytes 70215 (68.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 386 bytes 48848 (47.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2778 bytes 252345 (246.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2778 bytes 252345 (246.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



(5) 如果大家使用的是虚拟机，请把虚拟机系统的网卡（网络适配器）切换成桥接模式，如图 2-17 所示，然后重启虚拟机系统即可。



```
[root@localhost ~]# nmcli connection show
NAME      UUID                                  TYPE      DEVICE
home      8be600db-228b-42d2-a581-7f8893413efa ethernet  ens33
virbr0    e387a60c-8d58-477b-85be-7347adeea915 bridge    virbr0
company   ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a ethernet  --
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  --
[root@localhost ~]#
```



(6) 如果回到公司，可以停止 home 会话，启动 company 会话（连接）。

```
[root@RHEL7-1 ~]# nmcli connection down home
```

Connection 'home' successfully deactivated (D-Bus active path:

/org/freedesktop/NetworkManager/ActiveConnection/4)

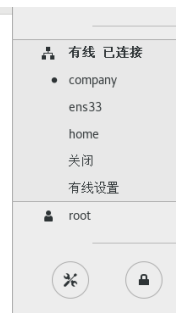
```
[root@RHEL7-1 ~]# nmcli connection up company
```

Connection successfully activated (D-Bus active path:

/org/freedesktop/NetworkManager/ActiveConnection/6)

```
[root@RHEL7-1 ~]# ifconfig
```

```
[root@localhost ~]# nmcli connection show
NAME      UUID                                  TYPE      DEVICE
home      8be600db-228b-42d2-a581-7f8893413efa ethernet  ens33
virbr0    e387a60c-8d58-477b-85be-7347adeea915 bridge    virbr0
company   ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a ethernet  --
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  --
[root@localhost ~]# nmcli connection down home
成功停用连接 "home" (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/1)
[root@localhost ~]# nmcli connection show
NAME      UUID                                  TYPE      DEVICE
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  ens33
virbr0    e387a60c-8d58-477b-85be-7347adeea915 bridge    virbr0
company   ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a ethernet  --
home      8be600db-228b-42d2-a581-7f8893413efa ethernet  --
[root@localhost ~]# nmcli connection up company
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/5)
[root@localhost ~]# nmcli connection show
NAME      UUID                                  TYPE      DEVICE
company   ac8b8bd6-894e-45ef-b1c5-aef75ae9f75a ethernet  ens33
virbr0    e387a60c-8d58-477b-85be-7347adeea915 bridge    virbr0
ens33     64d772b6-4d24-4534-9ce0-b8efd2e25664 ethernet  --
home      8be600db-228b-42d2-a581-7f8893413efa ethernet  --
```



(7) 如果要删除会话连接，请执行 nmcli 命令，执行“Edit a connection”命令，然后选中要删除的会话，按“Delete”按钮即可，如图 2-18 所示。

